Kadyrova Anel lab7

**Laboratory work 7**

1. How can we store large-object types?

Large objects (photos, videos, CAD files, etc.) are stored as a large object: • *blob*: binary large object -- object is a large collection of uninterpreted binary data (whose interpretation is left to an application outside of the database system)

• *clob*: character large object -- object is a large collection of character data

1. What is the difference between privilege, role and user?

grant <privilege list> on <relations> to <user or role>

* 1. create *accountant, administrator, support* roles and grant appropriate privileges

create role accountant;

grant select on transactions to accountant;

create role administrator;

grant insert, delete on accounts, customers, transactions to administrator;

create role support;

grant update on accounts, customers to support;

○ create some users and assign them roles,

create user Alex;

create user Jax;

grant support to Alex;

grant accountant to Jax;

○ give to some of them permission to grant roles to other users

grant select on transactions to Alex with grant option;

○ revoke some privilege from particular user

revoke select on transactions from Alex;

1. Add appropriate constraints
   1. - check if transaction has same currency for source and destination accounts (use assertion)

create assertion same\_currency check

(select ac2.currency,t.dst\_account, t.src\_account

from transactions t inner join accounts ac2 on t.dst\_account =ac2.account\_id

inner join accounts ac1 on t.src\_account=ac1.account\_id;)

○ add not null constraints

1. - Change currency column type to user-defined in accounts table

create domain Currency as varchar(3);

alter table accounts

alter column currency type varchar(3);

1. Create indexes:
   1. index so that each customer can only have one account of one currency

create index cust\_index on accounts(customer\_id, currency);

○ index for searching transactions by currency and balance

create index trans\_index on transactions(currency, balance);

create index src\_index on transactions(src\_account);

create index dst\_index on transactions(dst\_account);

1. Write a SQL transaction that illustrates money transaction from one account to another:
   1. create transaction with “init” status

○ increase balance for destination account and decrease for source account

○ if in source account balance becomes below limit, then make rollback

○ update transaction with appropriate status(commit or rollback)

begin;

INSERT INTO transactions VALUES (4, '2021-08-05 18:02:45.000000', 'RS88012', 'NT10204', 1000, 'init');

update accounts set balance=balance+1000 where account\_id='NT10204';

update accounts set balance=balance-1000 where account\_id='RS88012';

update transactions set status='commit' where id=4;

rollback;

update transactions set status='commit' where id=4;

commit;